

1/5/2

DIALOG(R) File 351:Derwent WPI

(c) 2004 Thomson Derwent. All rts. reserv.

010580660 \*\*Image available\*\*

WPI Acc No: 1996-077613/199608

XRPX Acc No: N96-064531

**Protecting shared system code and data in multi-tasking operating system  
- using synchronising mechanism for controlling access to shared system  
code and data by threads and requesting ownership of synchronising  
mechanism before cooperatively scheduled thread begins execution**

Patent Assignee: MICROSOFT CORP (MICT )

Inventor: CUTSHALL S M; SCHMIDT M A; THOMASON J G

Number of Countries: 018 Number of Patents: 006

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9600941	A1	19960111	WO 95US8287	A	19950630	199608 B
EP 715732	A1	19960612	EP 95926148	A	19950630	199628
			WO 95US8287	A	19950630	
JP 9502558	W	19970311	WO 95US8287	A	19950630	199720
			JP 96503459	A	19950630	
US 6148325	A	20001114	US 94268442	A	19940630	200060
EP 715732	B1	20030326	EP 95926148	A	19950630	200323
			WO 95US8287	A	19950630	
DE 69530048	E	20030430	DE 630048	A	19950630	200336
			EP 95926148	A	19950630	
			WO 95US8287	A	19950630	

Priority Applications (No Type Date): US 94268442 A 19940630

Cited Patents: 02Jnl.Ref; EP 381655; US 4945470

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 9600941 A1 E 26 G06F-009/46

Designated States (National): JP

Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL  
PT SE

EP 715732 A1 E 1 Based on patent WO 9600941

Designated States (Regional): DE FR GB

JP 9502558 W 29 Based on patent WO 9600941

US 6148325 A G06F-009/00

EP 715732 B1 E G06F-009/52 Based on patent WO 9600941

Designated States (Regional): DE FR GB

DE 69530048 E G06F-009/52 Based on patent EP 715732

Based on patent WO 9600941

Abstract (Basic): WO 9600941 A

The method for protecting shared code and data in a multi- tasking operating system (20) which includes a cooperative sub- system (24) having shared system code and data, and a pre-emptive sub-system (26) involves providing a synchronising mechanism for controlling access to the shared system code and data by threads, and providing a cooperatively scheduled thread to execute in the cooperative sub-system. Ownership of the synchronisation mechanism is requested before the cooperatively scheduled threads begins execution in the cooperative sub-system.

Ownership of the synchronisation mechanism must be requested and obtained before a preemptively scheduled thread can execute the shared system code in the cooperative sub-system. If the synchronisation mechanism is already owned, the requesting thread is blocked until ownership is released, otherwise the requesting thread is granted ownership. As no other thread can obtain ownership of the synchronising mechanism while one thread owns the synchronisation mechanism, the shared system code and data in the cooperative sub-system are protected.

USE/ADVANTAGE - Protecting shared code and data in multi-tasking operating system. Provides compatibility in operating system that

includes cooperative and preemptive sub-systems by protecting shared code and data in cooperative sub-system from preemptive capabilities of preemptive sub-system.

Dwg.2/4b

Title Terms: PROTECT; SHARE; SYSTEM; CODE; DATA; MULTI; OPERATE; SYSTEM;  
SYNCHRONISATION; MECHANISM; CONTROL; ACCESS; SHARE; SYSTEM; CODE; DATA;  
THREAD; REQUEST; SYNCHRONISATION; MECHANISM; COOPERATE; SCHEDULE; THREAD;  
BEGIN; EXECUTE

Derwent Class: T01

International Patent Class (Main): G06F-009/00; G06F-009/46; G06F-009/52

International Patent Class (Additional): G06F-009/40

File Segment: EPI

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表平9-502558

(43) 公表日 平成9年(1997)3月11日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 9/46

識別記号

3 4 0

庁内整理番号

9189-5B

F I

G 0 6 F 9/46

3 4 0 A

審査請求 未請求 予備審査請求 未請求(全 29 頁)

(21) 出願番号 特願平8-503459  
(86) (22) 出願日 平成7年(1995)6月30日  
(85) 翻訳文提出日 平成8年(1996)2月29日  
(86) 国際出願番号 PCT/US95/08287  
(87) 国際公開番号 WO96/00941  
(87) 国際公開日 平成8年(1996)1月11日  
(31) 優先権主張番号 08/268, 442  
(32) 優先日 1994年6月30日  
(33) 優先権主張国 米国 (US)  
(81) 指定国 EP(AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, M C, NL, PT, SE), JP

(71) 出願人 マイクロソフト コーポレイション  
アメリカ合衆国 ワシントン州 98052-  
6399 レッドモンド ワン マイクロソフ  
ト ウェイ (番地なし)  
(72) 発明者 シュミット マイケル エイ  
アメリカ合衆国 ワシントン州 98053  
レッドモンド トゥーハンドレッドアンド  
フォース アベニュー ノースイースト  
1646  
(74) 代理人 弁理士 中村 稔 (外7名)

最終頁に続く

(54) 【発明の名称】 マルチタスクオペレーティングシステムにおける共用コード及びデータを保護するための方法およびシステム

(57) 【要約】

マルチタスクオペレーティングシステムにおける共用コード及びデータ、特に、共用システムコード及びデータを保護する方法及びシステムが提供される。オペレーティングシステムは、協調型サブシステムと割り込み型サブシステムとを備える。協調型サブシステムは、共用システムコード及びデータを備える。この方法及びシステムは、スレッドによって共用システムコード及びデータへのアクセスを制御するための同期メカニズムを備える。同期メカニズムの所有権は、協調的に予定されたスレッドが協調型サブシステムにおいて実行できるようになる前に要求され、獲得されなければならない。さらに、同期メカニズムの所有権は、協調型サブシステムにおける共用システムコードを割り込み的に予定されたスレッドが実行可能になる前に要求され、獲得されなければならない。同期メカニズムが既に所有されている場合には、要求スレッドは、所有権が解放されるまでブロックされる。そうでなければ、要求スレッドが、所有権を許可されることになる。あるスレッドが同期メカニズムを所有している間は他のスレッドは同期メカニズムの所

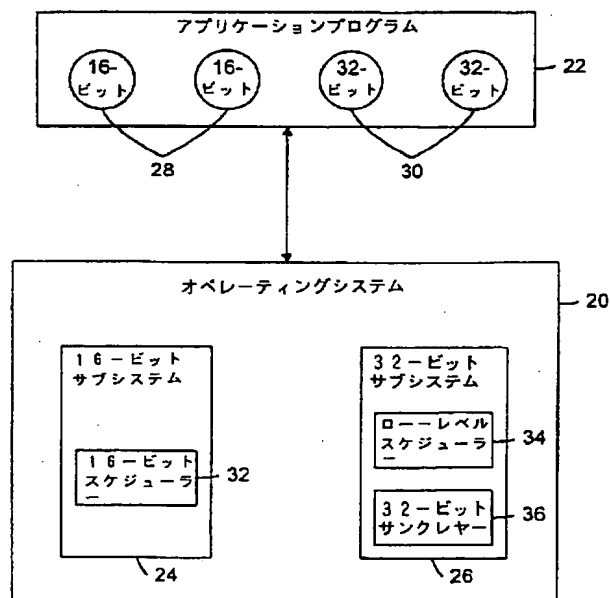


FIG. 2

**【特許請求の範囲】**

1. コンピュータシステムにおいて、協調型サブシステム及び割り込み型サブシステムを有するマルチタスクオペレーティングシステムにおける共用システムコード及びデータを保護する方法であって、前記協調型サブシステムが共用システムコード及びデータを備えており、

スレッドによって共用システムコード及びデータへのアクセスを制御するための同期メカニズムを提供し、及び

協調型サブシステムを実行するための協調的に予定されたスレッドを提供し、該協調的に予定されたスレッドが実行開始される前に同期メカニズムの所有権を要求するステップを有することを特徴とする方法。

2. コンピュータシステムにおいて、協調型サブシステム及び割り込み型サブシステムを有するマルチタスクオペレーティングシステムにおける共用システムコード及びデータを保護する方法であって、前記協調型サブシステムが共用システムコード及びデータを備えており、

スレッドによって共用システムコード及びデータへのアクセスを制御するための同期メカニズムを提供し、

協調型サブシステムを実行するための協調的に予定されたスレッドを提供し、該協調的に予定されたスレッドが実行開始される前に同期メカニズムの所有権を要求し、同期メカニズムがまだ所有されていない場合には、協調的に予定されたスレッドに対する同期メカニズムの所有権を許可し、

同期メカニズムの所有権が獲得されている場合には、前記協調的に予定されたスレッドが実行されることを許容し、及び

同期メカニズムの所有権が獲得されていない場合には、前記協調的に予定されたスレッドが実行されないようにブロックするステップを有することを特徴とする方法。

3. 同期メカニズムの所有権が獲得されていた場合には、前記協調的に予定されたスレッドが実行された後、同期メカニズムの所有権を解放するステップをさらに有することを特徴とする請求項2の方法。

4. 同期メカニズムを提供するステップが、スレッドによって共用システムコー

ド及びデータへのアクセスを制御するためのミューテックスを提供するステップを備えていることを特徴とする請求項2の方法。

5. 前記共用システムコード及びデータがオペレーティングシステムによって提供されるサービスをスレッドが要求できるアプリケーションプログラムインターフェイスを備え、該アプリケーションプログラムインターフェイスが1連のダイナミック・リンク・ライブラリとして実行されることを特徴とする請求項2の方法。

6. コンピュータシステムにおいて、協調型サブシステムと割り込み型サブシステムとを有するマルチタスクオペレーティングシステムにおける方法であって、

協調型サブシステムにおける共用システムコード及びデータを提供し、

前記共用システムコード及びデータが前記オペレーティングシステムによって提供されるサービスをスレッドが要求できるアプリケーション・プログラム・インターフェイスを備えており、該アプリケーション・プログラム・インターフェイスが一連のダイナミック・リンク・ライブラリとして実行されるものであり、該ダイナミック・リンク・ライブラリが前記オペレーティングシステムによって提供されるサービスを実行するためのコードを含んでおり、

スレッドによってダイナミック・リンク・ライブラリへのアクセスを制御するためのミューテックスの形態で同期メカニズムを提供し、

前記協調型サブシステムを実行するための協調的に予定されたスレッドを提供し、

該協調的に予定されたスレッドが実行開始される前にミューテックスの所有権を要求し、

ミューテックスがまだ所有されていない場合には、前記協調的に予定されたスレッドに対するミューテックスの所有権を許可し、

ミューテックスの所有権が獲得されている場合には、前記協調的に予定されたスレッドが実行されることを許容し、そして、

ミューテックスの所有権が獲得されていない場合には、前記協調的に予定されたスレッドが実行されないようにブロックするステップを備えたことを特徴とする方法。

7. コンピュータシステムにおいて、協調型サブシステム及び割り込み型サブシステムを有するマルチタスクオペレーティングシステムにおける共用システムコード及びデータを保護する方法であって、前記協調型サブシステムが共用システムコード及びデータを備えており、

スレッドによって共用システムコード及びデータへのアクセスを制御するための同期メカニズムを提供し、

協調型サブシステムにおいて共用システムコードを実行するための割り込み的に予定されたスレッドを提供し、そして、

前記割り込み的に予定されたスレッドが前記共用システムコードの実行を開始する前に前記同期メカニズムの所有権を要求するスレッドを有することを特徴とする方法。

8 コンピュータシステムにおいて、協調型サブシステム及び割り込み型サブシステムを有するマルチタスクオペレーティングシステムにおける共用システムコード及びデータを保護する方法であって、前記協調型サブシステムが共用システムコード及びデータを備えており、

スレッドによって共用システムコード及びデータへのアクセスを制御するための同期メカニズムを提供し、

協調型サブシステムにおいて共用システムコードを実行するための割り込み的に予定されたスレッドを提供し、

割り込み的に予定されたスレッドが共用システムコードの実行を開始する前に同期メカニズムの所有権を要求し、

同期メカニズムがまだ所有されていない場合には、前記割り込み的に予定されたスレッドに対する同期メカニズムの所有権を許可し、

同期メカニズムの所有権が獲得されていた場合には、前記割り込み的に予定されたスレッドが前記共用システムコードを実行することを許容し、

同期メカニズムの所有権が獲得されていない場合には、前記割り込み的に予定されたスレッドが共用システムコードを実行しないようにブロックするステップを備えた方法。

9. さらに、同期メカニズムの所有権が獲得されている場合には前記割り込み的

に予定されたスレッドが共用システムコードを実行した後前記同期メカニズムの所有権を解放するステップを備えていることを特徴とする方法。

10. 同期メカニズムを提供するステップがスレッドによって共用システムコード及びデータへのアクセスを制御するためのミューテックスを提供するステップを備えていることを特徴とする請求項8の方法。

11. 共用システムコード及びデータが、前記オペレーティングシステムによって提供されるサービスをスレッドが要求することができるアプリケーション・プログラム・インターフェイスを備えていることを特徴とする請求項8の方法。

12. コンピュータシステムにおいて、協調型サブシステムと割り込み型サブシステムとを有するマルチタスクオペレーティングシステムにおける方法であって、

前記協調型サブシステムにおいて共用システムコード及びデータを提供し、該共用システムコード及びデータは前記オペレーティングシステムによって提供されるサービスをスレッドが要求することかできるアプリケーション・プログラム・インターフェイスを備えており、該アプリケーション・プログラム・インターフェイスは、一連のダイナミック・リンク・ライブラリとして実行されるものであり、該ダイナミック・リンク・ライブラリは、オペレーティングシステムによって提供されるサービスを実行するためのコードを有しており、

スレッドによってダイナミック・リンク・ライブラリへのアクセスを制御するためのミューテックスの形態で同期メカニズムを提供し、

前記ダイナミック・リンク・ライブラリのコードを実行するために割り込み的に予定されたスレッドを提供し、

該割り込み的に予定されたスレッドが前記ダイナミック・リンク・ライブラリにおけるコードの実行を開始する前にミューテックスの所有権を要求し、

ミューテックスがまだ所有されていない場合には、前記割り込み的に予定されたスレッドに対するミューテックスの所有権を許可し、

ミューテックスの所有権が獲得されている場合には、割り込み的に予定されたスレッドのダイナミック・リンク・ライブラリのコードの実行を許容し、そして、

ミューテックスの所有権がまだ獲得されていない場合には、ダイナミック・

リンク・ライブラリのコードが実行されないように割り込み的に予定されたスレッドをブロックするステップを備えたことを特徴とする方法。

13. コンピュータシステムにおいて、協調型サブシステムと割り込み型サブシステムとを有するマルチタスクオペレーティングシステムにおける共用システムコード及びデータを保護する方法であって、

前記協調型サブシステムが前記共用システムコード及びデータを備えており

スレッドによって前記共用システムコード及びデータへのアクセスを制御するための同期メカニズムを提供し、

前記オペレーティングシステムにおいて実行するために協調的に予定されたスレッドを提供し、

前記協調型サブシステムにおける共用システムコードを実行するために割り込み的に予定されたスレッドを提供し、

前記協調的に予定されたスレッドが実行を開始される前に、前記同期メカニズムの所有権を要求し、

前記同期メカニズムがまだ所有されていない場合には、前記協調的に予定されたスレッドに対する同期メカニズムの所有権を許可し、

前記同期メカニズムの所有権が協調的に予定されたスレッドによって獲得されている場合には協調的に予定されたスレッドが実行されることを許容し、

同期メカニズムの所有権が協調的に予定されたスレッドによって獲得されていない場合には協調的に予定されたスレッドが実行されないようにブロックし、

前記割り込み的に予定されたスレッドが共用システムコードの実行を開始する前に同期メカニズムの所有権を要求し、

同期メカニズムがまだ所有されていない場合には、割り込み的に予定されたスレッドに対する同期メカニズムの所有権を許可し、

同期メカニズムの所有権が前記割り込み的に予定されたスレッドによって獲得されている場合には、割り込み的に予定されたスレッドが共用システムコードを実行することを許容し、そして、

同期メカニズムの所有権がまだ割り込み的に予定されたスレッドによって獲



得されていない場合には、割り込み的に予定されたスレッドが共用システムコ

ードを実行しないようにブロックするステップを備えたことを特徴とする方法。

14. さらに、同期メカニズムの所有権が協調的に予定されたスレッド獲得されている場合には、協調的に予定されたスレッドが実行された後に同期メカニズムの所有権を解放し、そして、

同期メカニズムの所有権が割り込み的に予定されたスレッドによって獲得されている場合には、割り込み的に予定されたスレッドが共用システムコードを実行した後同期メカニズムの所有権を解放するステップを備えたことを特徴とする請求項13の方法。

15. 同期メカニズムを提供するステップが、スレッドによって共用システムコード及びデータへのアクセスを制御するミューテックスを提供するステップを備えていることを特徴とする請求項13の方法。

16. 前記共用システムコード及びデータが、前記オペレーティングシステムによって提供されるサービスをスレッドが要求できるアプリケーション・プログラム・インターフェイスを備え、該アプリケーション・プログラム・インターフェイスが一連のダイナミック・リンク・ライブラリとして実行されることを特徴とする請求項13の方法。

17. コンピュータシステムにおいて、協調型サブシステムと割り込み型サブシステムを有するマルチタスクオペレーティングシステムにおける方法であって、前記協調型サブシステムにおける共用システムコード及びデータを提供し、該共用システムコード及びデータは前記オペレーティングシステムによって提供されるサービスをスレッドが要求することかできるアプリケーション・プログラム・インターフェイスを備えており、該アプリケーション・プログラム・インターフェイスは、一連のダイナミック・リンク・ライブラリとして実行されるものであり、該ダイナミック・リンク・ライブラリは、オペレーティングシステムによって提供されるサービスを実行するためのコードを有しており、

スレッドによってダイナミック・リンク・ライブラリへのアクセスを制御するためのミューテックスの形態で同期メカニズムを提供し、

前記ダイナミック・リンク・ライブラリのコードを実行するために割り込み的に予定されたスレッドを提供し、

前記ダイナミック・リンク・ライブラリにおけるコードを実行するために割り込み的に予定されたスレッドを提供し、

前記協調的に予定されたスレッドが実行を開始する前にミューテックスの所有権を要求し、

ミューテックスがまだ所有されていない場合には、前記協調的に予定されたスレッドに対するミューテックスの所有権を許可し、

ミューテックスの所有権が協調的に予定されたスレッドによって獲得されている場合には、協調的に予定されたスレッドの実行を許容し、

ミューテックスの所有権がまだ獲得されていない場合には、協調的に予定されたスレッドが実行されないようにブロックし、

割り込み的に予定されたスレッドが前記ダイナミック・リンク・ライブラリのコードの実行を開始する前にミューテックスの所有権を要求し、

前記ミューテックスがまだ所有されていない場合には割り込み的に予定されたスレッドに対するミューテックスの所有権を許可し、

ミューテックスの所有権が前記割り込み的に予定されたスレッドによって獲得されている場合には、割り込み的に予定されたスレッドが前記ダイナミック・リンク・ライブラリのコードを実行することを許容し、

そして、ミューテックスの所有権が割り込み的に予定されたスレッドによって獲得されていない場合には、割り込み的に予定されたスレッドがダイナミック・リンク・ライブラリのコードを実行しないようにブロックするステップを備えたことを特徴とする方法。

18. コンピュータシステムにおいて、協調型サブシステムと割り込み型サブシステムとを有するマルチタスクオペレーティングシステムにおける共用システムコード及びデータを保護するシステムであって、前記協調型サブシステムが共用システムコード及びデータとを備えており、

スレッドによって共用システムコード及びデータへのアクセスを制御する同

期メカニズムと、

協調型サブシステムにおいて実行するための協調的に予定されたスレッドと

前記協調型サブシステムにおける共用システムコードを実行するための割り

込み的に予定されたスレッドと、

協調的に予定されたスレッドの実行開始が許容される前に同期メカニズムの所有権を要求し、獲得しかつ協調的に予定されたスレッドが実行された後同期メカニズムの所有権を解放する協調型スケジューラーと、

割り込み的に予定されたスレッドが共用システムコードの実行を許容される前に同期メカニズムの所有権を要求し、獲得し、前記かつ割り込み的に予定されたスレッドが共用システムコードを実行した後同期メカニズムの所有権を解放するサンクレーヤーとを備えていることを特徴とするシステム。

19. 同期メカニズムがミューテックスであることを特徴とする請求項18のシステム。

20. 共用システムコード及びデータが、前記オペレーティングシステムによって提供されるサービスをスレッドが要求できるアプリケーション・プログラム・インターフェイスを備えており、該アプリケーション・プログラム・インターフェイスが一連のダイナミック・リンク・ライブラリとして実行されることを特徴とする請求項18記載のシステム。

21. コンピュータシステムにおいて、協調型サブシステムと割り込み型サブシステムとを有するマルチタスクオペレーティングシステムであって、

協調型サブシステムにおける共用システムコード及びデータであって、該共用システムコード及びデータは前記オペレーティングシステムによって提供されるサービスをスレッドが要求することができるアプリケーション・プログラム・インターフェイスを備えており、該アプリケーション・プログラム・インターフェイスは一連のダイナミック・リンク・ライブラリとして実行されるものであり、該ダイナミック・リンク・ライブラリは前記オペレーティングシステムによって提供されるサービスを実行するためのコードを備えており、

該システムは、さらに、

ダイナミック・リンク・ライブラリのコードを実行するスレッドによって前記ダイナミック・リンク・ライブラリへのアクセスを制御するためのミューテックスの形態の同期メカニズムと、

前記協調型サブシステムにおいて実行される協調的に予定されたスレッドと

前記ダイナミック・リンク・ライブラリのコードを実行するための割り込み的に予定されたスレッドと、

協調的に予定されたスレッドが実行開始を許容される前にミューテックスの所有権を要求して獲得するための協調型スケジューラーと、

前記割り込み的に予定されたスレッドがダイナミック・リンク・ライブラリのコードの実行開始を許容される前に、ミューテックスの所有権を要求し、獲得するサンクレーヤーとを備えたことを特徴とするシステム。

**【発明の詳細な説明】****マルチタスクオペレーティングシステムにおける共用コード  
及びデータを保護するための方法およびシステム****背景技術****1. 発明の分野**

本発明は、一般に、マルチタスクオペレーティングシステムに関し、特に、マルチタスクオペレーティングシステムの共用コード及びデータを保護するための方法及びシステムに関する。

**2. 関連技術の説明**

多くのオペレーティングシステムがマルチタスクを支持している。あるタスクは、独立したエンティティとして実行されるスタンドアローン・アプリケーションプログラムすなわちサブプログラムである。

マルチタスクは、コンピュータのシステムのコンポーネントの効率を改良するために、一度にメモリ内に1以上のタスクを持つ、現在実行しているいずれかのタスクに切替えるコンピュータの能力である。

マルチタスクオペレーティングシステムは、共用コード及びデータを保護するためのメカニズムを提供する。共用コードは、1つ以上のタスクによって実行することができるコードであり、共用データは、1つ以上のタスクによってアクセスすることができるデータである。マルチタスクオペレーティングシステムによって提供されるメカニズムは、コード共用及びデータ共用に関係する問題を防止する。コード共用及びデータ共用の問題は、1つのタスクが共用コードが処理する共用コード及びあるいはデータを変更し、その共用コードあるいは共用データが変更されたのち他のタスクが開始しあるいは継続してその共用コードを使用する場合に、発生する。コードあるいはそのデータの変更の後、開始し、あるいは継続して実行されるタスクはそのコード及びデータに対する変更によって意外な結果を生じせしめる可能性がある。

マルチタスクオペレーティングシステムの1つのタイプは、非割り込み型、すなわち、協調型、マルチタスクオペレーティングシステムである。協調型マルチ

タスクオペレーティングシステムでは、コンピュータシステムが適正に機能するために、タスク間の協調が必要となる。そのようなオペレーティングシステムについて、第1のタスクがCPUの制御に割りつけられると、他のすべてのタスクは、その第1のタスクがCPUの制御を放棄するまで、ブロック（たとえば、CPUの制御が得られなくなる）される。したがって、共用コード及びデータは、そのようなオペレーティングシステムで本来的に保護される。特に、CPUの制御は、いったんタスクが割りつけられると、タスクから離れることはできないので、あるタスクが共用コードあるいはそのデータを変更し、他のタスクが、その共用コードあるいはそのデータが変更された後に他のタスクが開始し、あるいは、その共用コードの実行を継続することは不可能である。CPUの制御に割りつけられたタスクはそのタスクが共用コードの実行を完了するまでCPUの制御を放棄しない。

他の形式のマルチタスクオペレーティングシステムは、割り込み型マルチタスクオペレーティングシステムである。協調型システムと異なり、割り込み型マルチタスクオペレーティングシステムは、コンピュータシステムが適正に機能するために、タスク間の協調は必要とならない。このようなオペレーティングシステムに関し、システムは、1つのタスクに対してCPUの制御を割りつけ、その後特定の時間が経過するか、特定の事象が発生したときそのタスクから制御を引き上げる。したがって、共用コードと共用データは、そのようなオペレーティングシステムに、本来的に保護されているものではない。例えば、あるタスクは、オペレーティングシステムが当該タスクからCPUの制御を引き上げるときに共用コードあるいはそのデータを変更することができる。この結果、共用コードあるいはそのデータが変更された後において、他のタスクがCPUの制御に割りつけられ、その共用コードの実行を開始あるいは継続することができる。この結果、この状況によって、上記したコード共用及びデータ共用問題が生じる可能性がある。

#### 発明の概要

本発明の1つの特徴は、共用コード及びデータを保護するための方法及びシス

テムを提供することであり、特に、マルチタスクオペレーティングシステムにおける共用システムコード及びデータを保護するための方法を及びシステムを提供することである。このオペレーティングシステムは、協調型サブシステムと割り込み型サブシステムとを備えている。協調型サブシステムは、共用システムコード及びデータを含んでいる。この方法及びシステムは、スレッドによって共用システムコード及びデータにアクセスするように制御するための同期メカニズムを備えている。同期メカニズムの所有権は、協調的に予定されたスレッドが協調型サブシステムで実行することができるようになる前に要求され、かつ獲得されなければならない。さらに、同期メカニズムの所有権は、割り込み的に予定されたスレッドが協調型サブシステムにおける共用システムコードを実行することができるようになる前に、要求され、獲得されなければならない。同期メカニズムが既に所有されている場合には、所有権が解放されるまで要求スレッドはブロックされる。そうでなければ、要求スレッドは所有権を許可されることになる。他のいずれのスレッドも、あるスレッドが同期メカニズムを所有している間は、同期メカニズムの所有権を得ることができないので、協調型サブシステムにおける共用システムコード及びデータは保護される。

#### 図面の簡単な説明

図1は、本発明の好ましい実施例が動作するコンピュータシステムのコンポーネントを図示するブロックダイヤグラムであり、

図2は、図1のオペレーティングシステムのコンポーネントをさらに詳細に図示したブロックダイヤグラムであり、

図3は、本発明の好ましい実施例にしたがって16-ビットスレッドがミューテックスの所有権を要求し、獲得し、そして解放するかを図示したハイレベルフローチャートであり、

図4及び図5は、本発明の好ましい実施例にしたがって32-ビットスレッドがミューテックスの所有権を要求し、獲得し、そして解放するかを図示したハイレベルフローチャートである。

#### 好ましい実施例の詳細な説明

本発明の好ましい実施例は、共用コード及びデータ、特に、マルチタスクオペ

レーティングシステムにおける共用システムコード及びデータを保護する方法とシステムを提供するものである。

本発明の方法及びシステムは、マルチタスクオペレーティングシステムにおける共用システムコード及びデータを保護する同期メカニズムを備えている。オペレーティングシステムは、協調型サブシステム及び割り込み型サブシステムを備えている。協調型サブシステムは、共用システムコード及びデータを含んでいる。後述するように同期メカニズムの所有権は、協調的に予定されたスレッドが協調型サブシステムにおいて実行可能になる前に要求され、獲得されなければならない。さらに、同期メカニズムの所有権は、割り込み的に予定されたスレッドが協調型サブシステムにおいて共用システムコードを実行できるようになる前に要求され、獲得されなければならない。本発明の同期メカニズムは、協調的サブシステムにおける共用コード及びデータを、割り込み型サブシステムの割り込み可能性から保護することによって協調型サブシステム及び割り込み型サブシステムの両方を含むオペレーティングシステムにおける両立性を提供する。

内部において本発明の好ましい実施例が動作するコンピュータシステム10が図1に図示されている。コンピュータシステム10は、中央演算ユニット(CPU)12、主記憶装置14、第2記憶装置16及び入出力(I/O)装置18を備えている。オペレーティングシステム20及びアプリケーションプログラム22は、第2記憶装置16にストアされており、CPU12の実行のために主記憶装置14にロードされる。アプリケーションプログラム22のように、主記憶装置14にロードされ、CPU12によって実行のために用意された、プログラムは、プロセスとよばれる。プロセスは、コード、データ及びプログラムに属する他のリソースを有する。プロセスにおける実行パスは、スレッドとよばれる。スレッドは、一連の実行命令、関係するCPUレジスタ値、及びスタックを有する。プロセスは、少なくとも1つのスレッドを有する。マルチスレッドオペレーティングシステムでは、プロセスは、1つ以上のスレッドを持つことができる。スレッドは、CPU12の制御を受け入れるエンティティである。

オペレーティングシステム20のコンポーネントは、図2に示されている。本発明の好ましい実施例では、オペレーティングシステム20は、協調型マルチタ



スクオペレーティングサブシステム24及び割り込み型マルチタスクサブシステム26を備えている。さらに、協調型マルチタスクサブシステム24は、16ビットシステム（以下16ビットサブシステムという）であり、割り込み型マルチタスクサブシステム26は32ビットシステム（以下32ビットサブシステムという）である。したがって、アプリケーションプログラム22は、協調的に予定された16ビットアプリケーション28と割り込み的に予定された32ビットアプリケーション30の両方を備えている。オペレーティングシステム20は、“マイクロソフトウィンドウズ”バージョン3.1の改良バージョンであり、ワシントン州レッドモンドのマイクロソフトコーポレーションによって販売されているオペレーティングシステムである。ウィンドウズ3.1は、協調型マルチタスクオペレーティングシステムである。本発明の好ましい実施例におけるオペレーティングシステム20は、ウィンドウズ3.1によって提供される協調型マルチタスクサブシステムに対する改良として、割り込み型マルチタスクサブシステムを提供するものである。このような改良バージョンは、早い時期の協調型システムに対して書かれたアプリケーションと、最近の割り込み型システムように書かれたアプリケーションとの両立を与えるものである。

本発明は、ウィンドウズ3.1の改良バージョンに関して記載されたものであるが、当業者は、他のオペレーティングシステムが本発明を実施するのに使用できることを理解するであろう。ウィンドウズ3.1の改良バージョンの選択は、単に例示的なものである。

16ビットサブシステム24は、16ビットアプリケーション28が、オペレーティングシステム20によって与えられたローレベルサービスを要求し、実行することができる16ビットアプリケーションプログラムインターフェース（API）備えている。同様に、32ビットサブシステム26は、32ビットアプリケーション30が、オペレーティングシステム20によって与えられたローレベルサービスを要求し、実行することができる32ビットAPIを備えている。さらに、16ビットサブシステムは、16ビットアプリケーション28の協調的スケジュールを処理することを担当する16ビットスケジューラー32を備えるとともに、32ビットサブシステム26は、32ビ

ットアプリケーション30の割り込み型スケジューリングを処理するのを担当するローレベルスケジューラー34を備えている。そのような16ビットスケジューラーは、ワシントン州、レッドモンドのマイクロソフト・コーポレーションによって販売されているマイクロソフト・ウインドウズバージョン3.1、オペレーティングシステムによって与えられるものであり、そのようなローレベルスケジューラーはマイクロソフト・ウインドウズ NTオペレーティングシステムによって与えられるものである。

16ビットアプリケーション28が16ビットAPIファンクションをコールすると、そのコールは、16ビットサブシステム24によって直接取り扱われる。しかし、32ビットアプリケーション30が32ビットAPIファンクションをコールした場合には、そのコールは、通常は、32ビットサブシステム26によっては直接取り扱われない。むしろ、32ビットAPIファンクションと同じ目的をもつ16ビットAPIファンクションが存在する場合には、その32ビットAPIファンクションに対するコールは、通常は、同等な16ビットAPIファンクションに対するコールに変換され、16ビットサブシステム24によって取り扱われる。しかし、32ビットAPIファンクションと同じ目的をもつ16ビットAPIファンクションが存在しない場合には、あるいは、32ビットAPIファンクションが32ビットサブシステム26によって効率的に処理することができるならば、その32ビットAPIファンクションに対するコールは32ビットサブシステム26によって直接処理される。32ビットAPIファンクションに対するコールを同等な16ビットAPIファンクションに対するコールに変換し、つぎに、16ビットAPIファンクションの結果を32ビットフォーマットに再変換するプロセスは、“サンキング(thunking)”として知られている。このプロセスは、32ビットサンク(thunk)レヤー36で発生するものであって、16ビットフォーマットに、及びこれからの変換を行なうコードである。このサンクレヤーはマイクロソフト・ウインドウズ NTオペレーティングシステムによって提供される。

16ビットAPIと32ビットAPIの両方が1連のダイナミック・リン

ク・ライブラリ (DDL's) として実行される。DDLは、さまざまなファンクションを実行する実行可能コードを含むライブラリモジュールである。各ファンクションについてのコードは、DDL内にストアされている。アプリケーション・プログラムがDDL内のファンクションへのコールを含む場合には、DDLは、実行時においてそのアプリケーション・プログラムにダイナミックリンクされる。アプリケーション・プログラムは、そのファンクションがアプリケーション・プログラムのコードの一部であるかのようにDDL内におけるファンクションを使用するものである。

本発明のAPI'sを形成するDDL'sは、リエントラントではない。リエントラントであるコードは実行時中に修正不可能であり、幾つかのスレッドによって享有することができるよう書かれている。あるスレッドがリエントラントコードを実行しており、かつ他のスレッドがその第1のスレッドの実行をインタラプトしているとき、第2のスレッドがなんらのコード共用あるいはデータ共用の問題なしに同じコードの実行を開始し、あるいは継続することができる。したがって、DDLがリエントラントであるならば、1つ以上のスレッドが単一のDDLを使用するプロセス中に存在することとなる。DDLがリエントラントでない場合に生じる問題は、コードが変更可能であり、DDLにある同じコード及びデータがDDLを使用するすべてのスレッドの中で共用されるということである。この結果、1つのスレッドが非リエントラントDDL中のコードを実行しており、他のスレッドが第1のスレッドの実行をインタラプトし、DDL中の同じコードの実行を開始し、あるいは継続しようとしている場合には、コード共用あるいはデータ共用の問題が生じる可能性がある。本発明の16-ビットAPIを形成するDDLは、非リエントラントであり、オペレーティングシステム20が協調型16-ビットサブシステム24及び割り込み型32-ビットサブシステム26を備えているので、第1のスレッドが他の16-ビットAPIファンクションの実行を完了する前に、メカニズムは、第2のスレッドが16-ビットAPIファンクションを実行しないようにする必要がある。

16-ビットAPI中の共用コード及びデータを保護するための、本発明によって提供される方法及びシステムは、ミューテックス (mutex) の形態の同

期メカニズムを備えている。当業者は、本発明が、他の同期メカニズム、例えば、クリティカル・セクションあるいはセマフォなどを同様に使用することができること理解するであろう。ミューテックスは、共用リソースの同時使用を防止するのに使用できるデータストラクチャである。この場合、共用リソースは、16ビットAPIのコード及びデータである。スレッドが共用リソースの使用を望む場合には、スレッドは、そのミューテックスの所有権を要求する。もしミューテックスがすでに他のスレッドで所有されている場合には、要求スレッドは、所有権が解放されるまでブロックされる。そうでなければ、要求スレッドは、所有権が許容されるからである。本発明の好ましい実施例では、スレッドがミューテックスの所有権を要求し、獲得し、そして解放する時間と方法は、16ビットスレッドと32ビットスレッドとで異なる。

16ビットスレッドに対しては、ミューテックスの所有権は、16ビットスレッドがそのアプリケーションコードをコールイン（呼び込む）する前に要求され、獲得されなければならない。図3は、16ビットスレッドがミューテックスの所有権を要求し、獲得し、そして解放するように実行されるステップを示している。まず、16ビットアプリケーション28がスタートすると、16ビットステップS1が作られ、16ビットスケジューラー32にリンクされる（ステップS38）。16ビットスレッドがそのアプリケーションコードにコールインする前にその16ビットスレッド16ビットスケジューラー32の中でミューテックスの所有権を要求する（ステップS40）。ミューテックスがすでに所有されていた場合には（ステップS42）、16ビットスレッドは、所有権が解放され、そのスレッドが実行の優先権を有するまでブロックされる（ステップS44）。そうでなければ、16ビットスレッドは所有権を許されるからであり（ステップS46）、そのアプリケーションコードをコールインして実行するからである（ステップS48）。16ビットスレッドがCPU12の16ビットスケジューラー32に対する制御を放棄したとき、16ビットスレッドは、16ビットスケジューラー32の内のミューテックスの所有権を解放する（ステップS50）。16ビットスレッドがミューテックスを所有している間、16ビットスレッドは、16ビットAPI内のファンクションに

対して多重コールを行なうことができる。16-ビットスレッドがロットを所有している間、他のスレッドは、ミューテックスの所有権を獲得することができないので、16-ビットAPI内のコードとデータは保護される。

32-ビットスレッドについては、(同等の32-ビットAPIファンクションへのコールからサンク(thunk)された16-ビットAPIファンクションへのコールが実行を始める前に)要求され、獲得されなければならない。図4及び図5は、ミューテックスの所有権を要求し、獲得しそして解放するために32-ビットスレッドがおこなうべきステップを集合的に図示したものである。まず、32-ビットアプリケーション30がスタートし、32-ビットが作られローレベルスケジューラー34(ステップS52)にリンクされる。32-ビットスレッドがCPU12の制御に割りつけられると、32-ビットスレッドがそのアプリケーションコードにコールインされ、実行される(ステップ54)。実行中、32-ビットスレッドがサンクしなければならない(すなわち、同等の16-ビットAPIファンクションへのコールに変換しなければならない)32-ビットAPIファンクションをコールしたときには(ステップ56)、32-ビットAPIファンクションは、その32-ビットサンクレヤー36に制御を移行する(ステップ58)。32-ビットサンクレヤー36は32-ビットAPIファンクションへのコールを同等な16-ビットAPIファンクションへのコールに変換し(ステップ60)、32-ビットスレッドは、32-ビットサンクレヤー36内のミューテックスの所有権を要求する(ステップ62)。もし、ミューテックスがすでに所有されている場合には(ステップ64)、32-ビットスレッドは、所有権が解放され、スレッドが実行の優先権をもつまでブロックされる(ステップ66)。そうしなければ、32-ビットスレッドは、所有権を許容され(ステップ68)、同等の16-ビットAPIファンクションをコールすることになる(ステップ70)。16-ビットAPIファンクションはそのコールを処理し、その結果を32-ビットサンクレヤー36に戻し、32-ビットスレッドは、32-ビットサンクレヤー36内のミューテックスの所有権を解放する(ステップ74)。最後に、32-ビットサンクレヤー36は、16-ビットAPIファンクションからの結果を32-ビットフォーマットに再変換し、制御を

32-ビットスレッドに戻す(ステップ76)。上記したように、32-ビットスレッドがロックを所有しているあいだ他のスレッドは、ミューテックスの所有権を獲得することはできないので、16-ビットAPIのコードとデータは保護される。

上記の32-ビットスレッドに関連する議論において、ミューテックスの所有権は、32-ビットAPIファンクションへのコールが同等の16-ビットAPIファンクションへのコールに変換された後要求され、16-ビットAPIファンクションからの結果が32-ビットフォーマットに再変換される前に、解放される。当業者は、ミューテックスの所有権が、32-ビットAPIファンクションへのコールが同等の16-ビットAPIファンクションへのコールに変換される前に要求でき、16-ビットAPIファンクションからの結果が32-ビットフォーマットに再変換される前に解放することもできることを理解するであろう。

当業者は、本発明が共用コード及びデータ、特に、マルチタスクオペレーティングシステムにおける共用システムコード及びデータにおけるを保護するための方法及びシステムを提供するものであることを理解するであろう。この方法及びシステムは、マルチタスクオペレーティングシステムの共用システムコード及びデータを保護する同期メカニズムを備えている。オペレーティングシステムは、協調型サブシステム及び割り込み型サブシステムを備えている。協調型サブシステムは、共用システムコード及びデータを備えている。同期メカニズムの所有権は、協調的に予定されたスレッドが協調型サブシステムにおいて実行することかできるようになる前に要求され、獲得されなければならない。さらに、同期メカニズムの所有権が、割り込み的に予定されたスレッドが、協調型サブシステムにおいて共用システムコードを実行できるようになる前に、要求され、獲得されなければならない。本発明の同期メカニズムは、割り込み型サブシステムの割り込み的な能力から協調型サブシステムにおける共用コード及びデータを保護することによって協調型サブシステム及び割り込み型サブシステムの両方を備えたオペレーティングシステムの両立性を与えるものである。本発明は、実施例に関連して図示され、記載されているが、本明細書を読み、理解することによって当業者は

、同等な修正、変更を行なうであろう。本発明は、そのようなすべての同等な修正、

変更を包含するものであり、以下の特許請求の範囲によってのみ限定されるものである。

【図1】

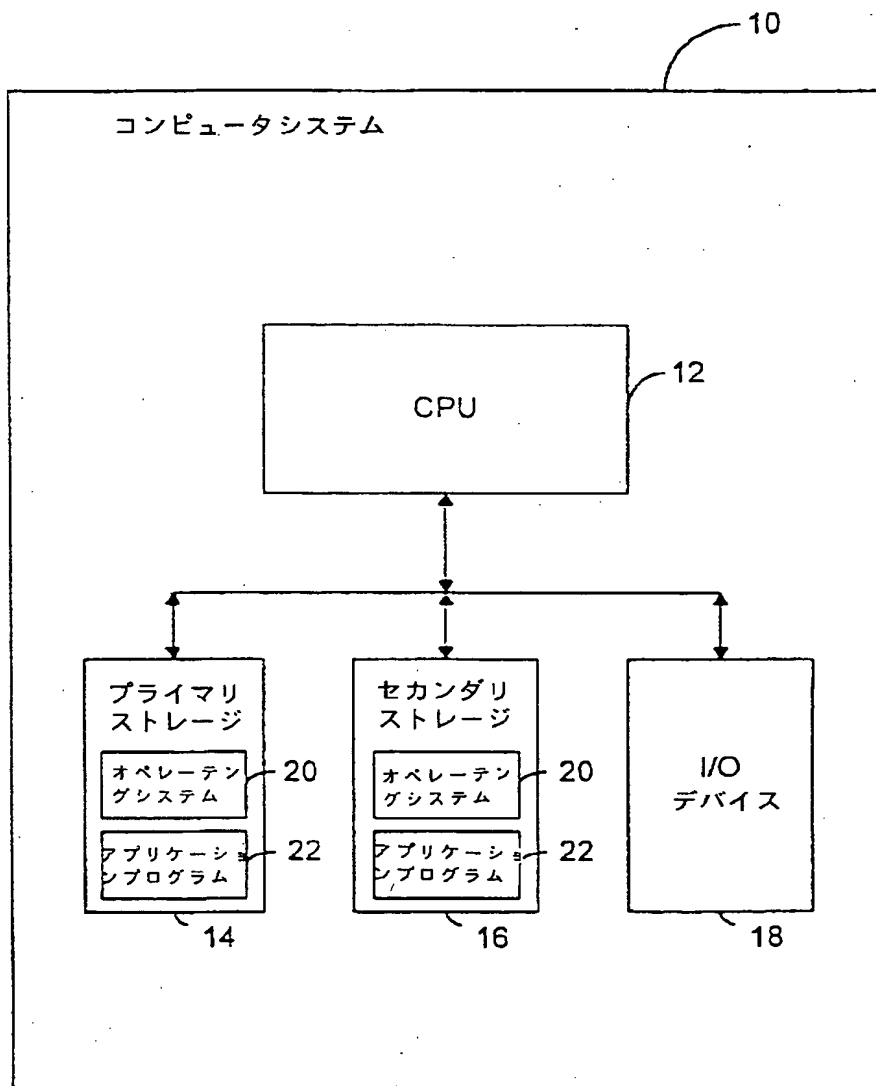


FIG. 1

【図2】

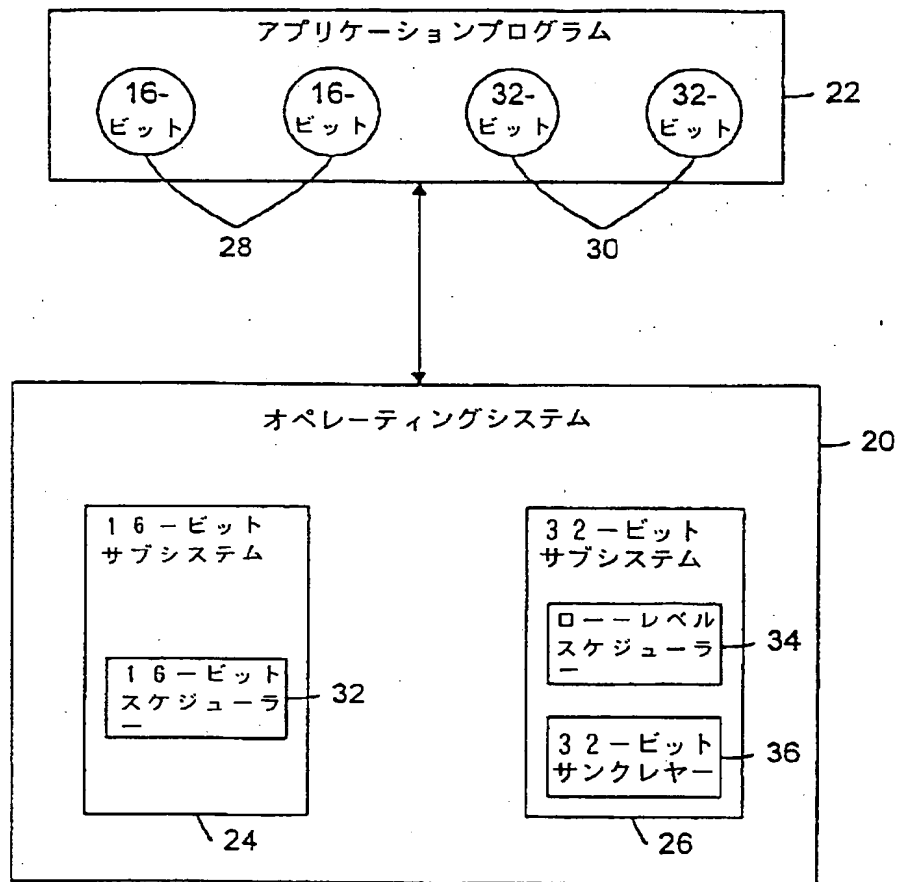


FIG. 2



【図3】

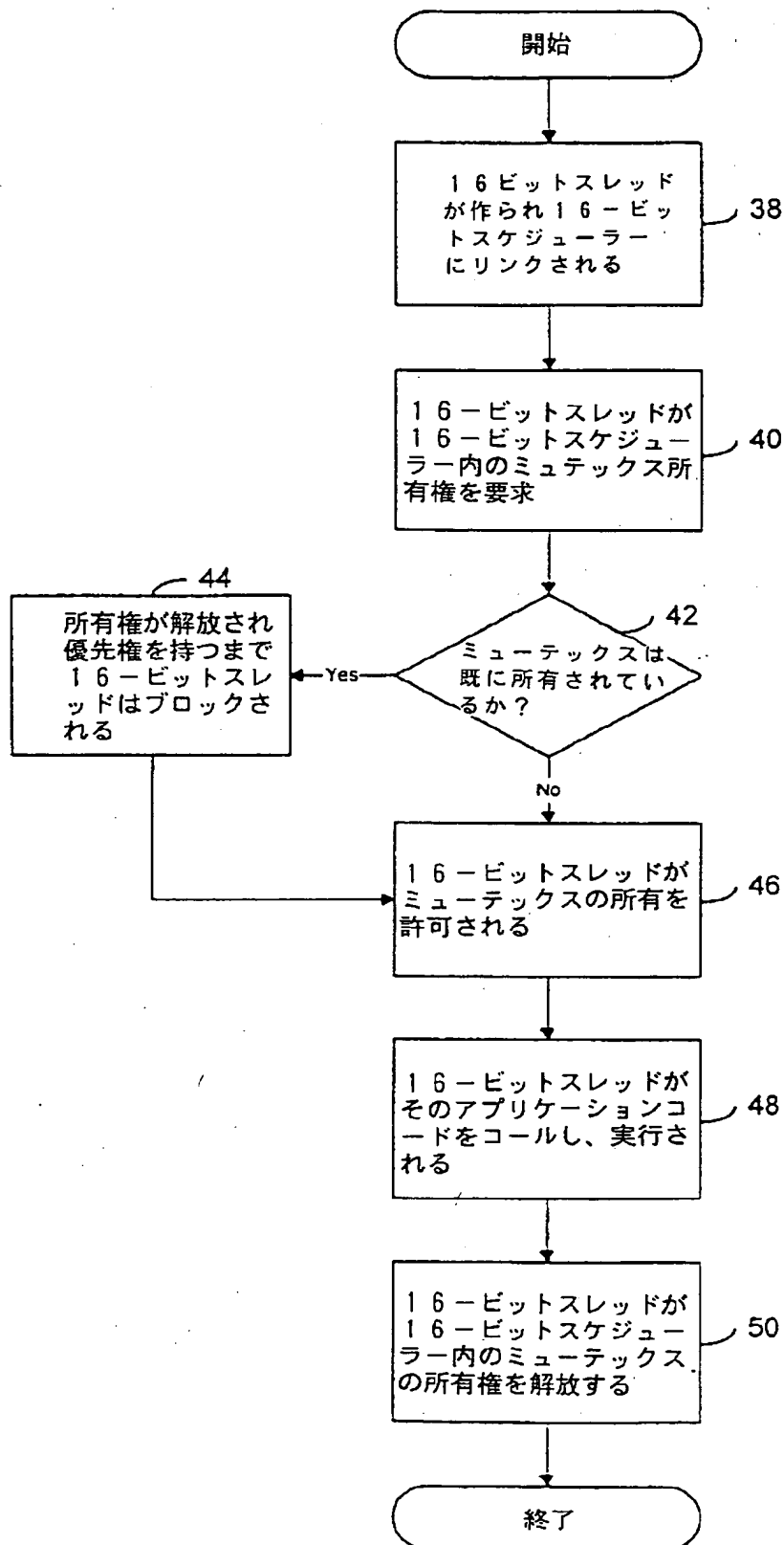


FIG. 3

【図4】

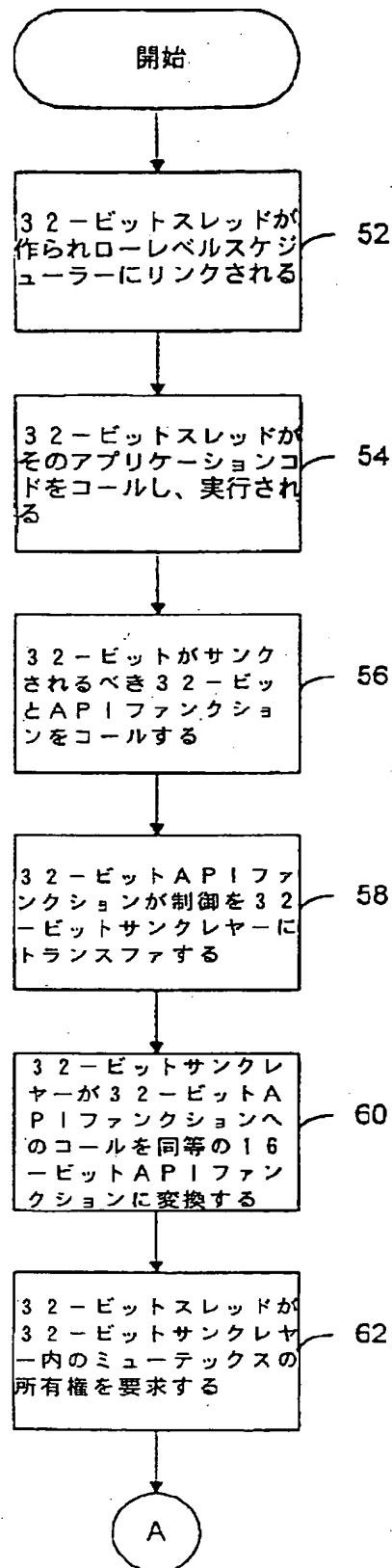


FIG. 4A

【図4】

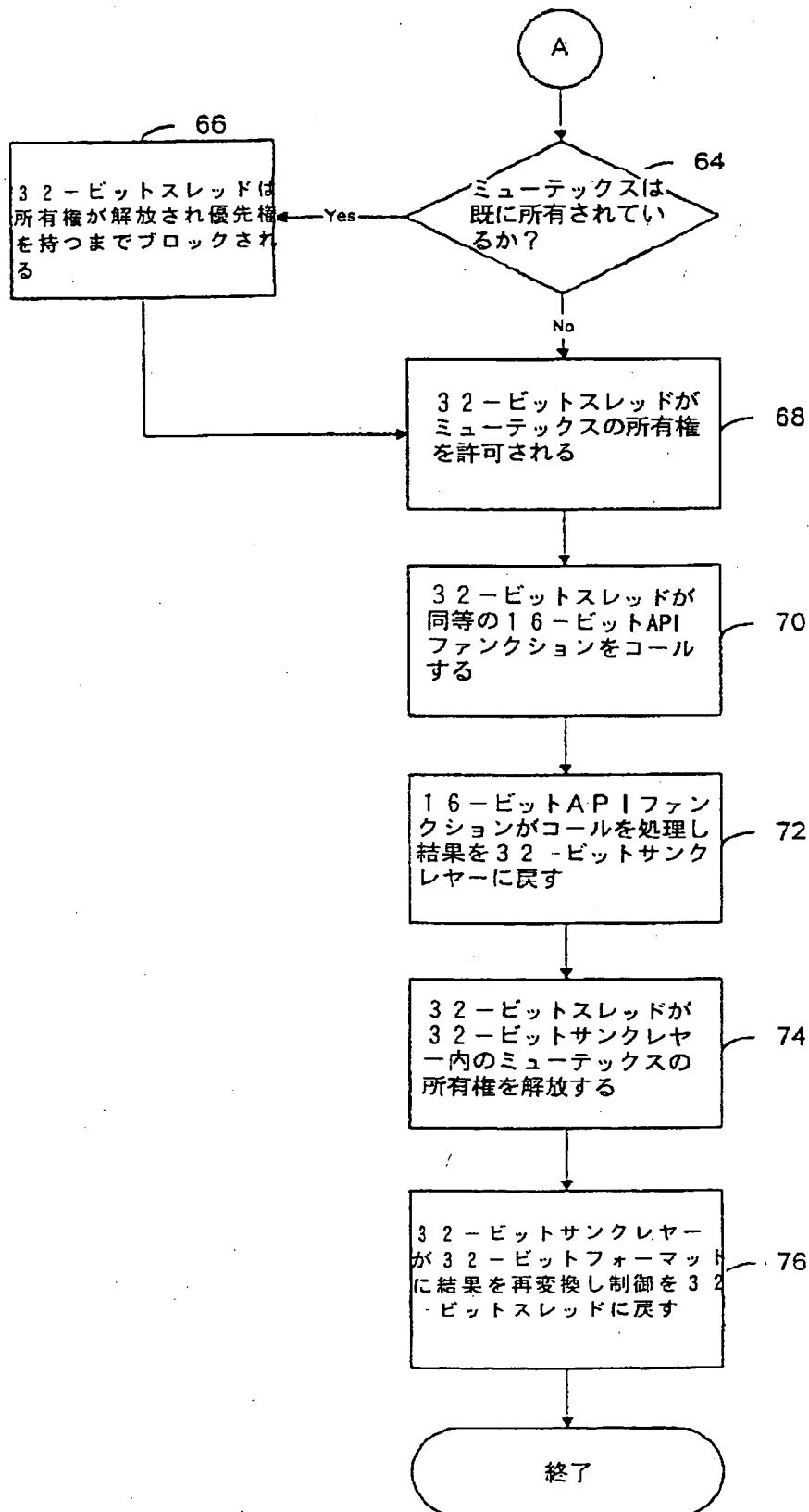


FIG. 4B

【国際調査報告】

## INTERNATIONAL SEARCH REPORT

International Application No. <b>PCT/US 95/08287</b>	
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> <b>IPC 6 G06F9/46</b>	
According to International Patent Classification (IPC) or to both national classification and IPC	
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) <b>IPC 6 G06F</b>	
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched	
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)	
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>	
Category	Citation of document, with indication, where appropriate, of the relevant passages
X	<b>REAL TIME SYSTEMS,</b> vol. 3, no. 2, 1 May 1991, NL, pages 149-163, XP 000306470 <b>M. BOTTAZZI ET AL.: 'A Hierarchical</b> <b>Approach to Systems with Heterogeneous</b> <b>Real-Time Requirements'</b>
Y	see page 150, line 15 - line 26 see page 156, line 28 - page 158, line 4 see page 152, line 31 - page 153, line 8 --- -/-
	1-4, 7-10, 13-15   5,6,11, 12,16-21
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.	
<input checked="" type="checkbox"/> Patent family members are listed in annex.	
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. "A" document member of the same patent family	
Date of the actual completion of the international search  <b>11 October 1995</b>	Date of making of the international search report  <b>10.11.95</b>
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	Authorized officer  <b>Wiltink, J</b>

## INTERNATIONAL SEARCH REPORT

Int. Patent Application No.

PCT/US 95/08287

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 34, no. 4B, September 1991, NEW YORK, US, pages 314-317, 'Sixteen to Thirty-two Bit Operating System Compatibility Method for Personal Computers.' see the whole document -----	5,6,11, 12,16-21
A	US,A,4 945 470 (TAKAHASHI) 31 July 1990  see abstract; figure 1 see column 1, line 29 - line 51 see column 1, line 66 - column 2, line 47 -----	1,6,7, 12,13, 18,21
A	EP,A,0 381 655 (IBM) 8 August 1990  see abstract; claim 1 see page 4, line 1 - line 31 -----	1,6,7, 12,13, 18,21

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International Application No.

**PCT/US 95/08287**

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US-A-4945470	31-07-90	JP-A- 61006741	13-01-86
EP-A-381655	08-08-90	JP-C- 1856975	07-07-94
		JP-A- 2231640	13-09-90
		US-A- 5319782	07-06-94

---

フロントページの続き

(72)発明者 トマソン ジョナサン ジー  
アメリカ合衆国 ワシントン州 98053  
レッドモンド トゥーハンドレッドアンド  
サーティフィフス プレイス ノースイ  
ースト 7845

(72)発明者 カッツホール スコット エム  
アメリカ合衆国 ワシントン州 98014  
カーネーション トゥーハンドレッドアン  
ドエイティナインス アベニュー ノース  
イースト 816

## 【要約の続き】

有権を獲得することはできないので、協調型サブシステ  
ムにおける共用システムコード及びデータは、保護され  
る。